



37

Ao construir um protótipo que calculava o risco sanitário de um terreno, um desenvolvedor principiante começou a desenhar um modelo relacional, o que o levou a propor as seguintes tabelas (onde o sublinhado representa a chave primária):

```
Terreno(rua, numero, tipoTerreno, CNPJ, nomeEmpresa, codigoRisco, nomeRisco, probabilidadeRisco)
Empresa(CNPJ, nomeEmpresa, CPFs, nomePessoas)
```

Ao revisar o trabalho, um segundo desenvolvedor, mais experiente, detectou as seguintes propriedades:

1. $\text{codigoRisco} \rightarrow \text{nomeRisco}$
2. $\text{CNPJ} \rightarrow \text{nomeEmpresa}$
3. $\text{CPF} \rightarrow \text{nomePessoa}$
4. $\{\text{rua}, \text{numero}\} \rightarrow \text{CNPJ}$
5. Uma empresa possui vários donos, cada um com um CPF único, e um dono pode possuir várias empresas
6. Um terreno pode possuir vários riscos

Esse segundo desenvolvedor decidiu, então, colocar as tabelas na terceira forma normal, usando o número mínimo de tabelas.

O esquema que possui as seis propriedades detectadas pelo segundo desenvolvedor e que representa todas as informações desejadas pelo primeiro desenvolvedor é

- (A) Terreno(rua, numero, tipoTerreno, CNPJ)
Empresa(CNPJ, nomeEmpresa)
Dono(CNPJ, CPF)
Risco(rua, nome, codigoRisco, probabilidadeRisco)
TipoRisco(codigoRisco, nomeRisco)
PessoaFisica(CPF, nome)
- (B) Terreno(rua, numero, tipoTerreno, CNPJ, codigoRisco, probabilidadeRisco)
Empresa(CNPJ, nomeEmpresa)
Dono(CNPJ, CPF)
TipoRisco(codigoRisco, nomeRisco)
PessoaFisica(CPF, nome)
- (C) Terreno(rua, numero, tipoTerreno, CNPJ)
Empresa(CNPJ, nomeEmpresa)
Risco(rua, nome, codigoRisco, probabilidadeRisco)
TipoRisco(codigoRisco, nomeRisco)
PessoaFisica(CPF, nome, CNPJ)
- (D) Terreno(rua, numero, tipoTerreno, CNPJ, nomeEmpresa, codigoRisco, nomeRisco, probabilidadeRisco)
Empresa(CNPJ, CPF, nomeEmpresa, nomePessoa)
- (E) Terreno(rua, numero, tipoTerreno, CNPJ, nomeEmpresa, codigoRisco, probabilidadeRisco)
Risco(codigoRisco, nomeRisco)
Empresa(CNPJ, CPF, nomeEmpresa, nomePessoa)



38

Para gerar um gráfico de dispersão, um programador precisava consultar duas tabelas, T1 e T2. Ele decidiu, então, usar um LEFT JOIN, como em

```
SELECT * FROM T1 LEFT JOIN T2 USING (CHAVE);
```

Essa consulta resultou em 214 linhas.

Por motivos de segurança, ele fez outra consulta semelhante, apenas trocando o LEFT JOIN por um JOIN, e essa segunda consulta resultou em 190 linhas.

O que pode explicar corretamente a quantidade diferente de linhas nas consultas realizadas?

- (A) CHAVE é a chave primária de T1, mas apenas um campo da chave primária de T2.
(B) CHAVE é a chave primária de T2, mas apenas um campo da chave primária de T1.
(C) T1 possui linhas cujo valor de CHAVE não está presente na T2.
(D) T2 possui linhas cujo valor de CHAVE não está presente na T1.
(E) T2 possui linhas com todas as chaves presentes em T1, mas com campos nulos.

**40**

Ao coletar dados em um sistema compatível com SQL 2008 para fazer uma análise de dados, um programador percebeu que havia dois campos, `data_de_nascimento` e `data_de_emissão_RG`, em que o valor de `data_de_emissão_RG` sempre deve ser mais recente que `data_de_nascimento`. Percebeu, porém, que em 10% das linhas acontecia o inverso, isto é, `data_de_nascimento` era mais recente que `data_de_emissão_RG`. Ele corrigiu os dados nessas linhas, verificando que estavam consistentemente trocados, mas, preocupado que tal problema voltasse a acontecer, resolveu solicitar ao DBA uma alteração da tabela, de forma que `data_de_emissão_RG` sempre tivesse que ser mais recente que `data_de_nascimento`.

O DBA atendeu adequadamente a esse pedido do programador por meio de uma restrição em SQL 2008 do tipo

- (A) CHECK
- (B) INSPECT
- (C) TEST
- (D) VALIDATE
- (E) VERIFY

**45**

Após um treinamento em SQL padrão 2008, compatível com ambiente MS SQL Server 2008, um escriturário do Banco Z precisou utilizar os conhecimentos adquiridos para criar uma tabela no sistema de banco de dados desse Banco. A tabela a ser criada é de fornecedores, e tem os seguintes campos: CNPJ, nome do fornecedor e país de origem. As características gerais da tabela são:

- o campo CNPJ é chave primária e contém 14 caracteres, sendo que os caracteres devem se ater aos numéricos ["0" a "9"], e o caractere zero "0" não pode ser ignorado, seja qual for a posição dele (início, meio ou fim da chave);
- o campo NOME contém 20 caracteres e aceita valor nulo;
- o campo PAIS contém 15 caracteres e não aceita valor nulo.

Nesse contexto, o comando SQL2008 que cria uma tabela com as características descritas acima é

- (A)

```
CREATE TABLE Fornecedores
(CNPJ INTEGER PRIMARY KEY,
NOME VARCHAR(20) ACCEPT NULL,
PAIS VARCHAR(15) NOT NULL)
```
- (B)

```
CREATE TABLE Fornecedores
(CNPJ CHAR(14) PRIMARY KEY,
NOME VARCHAR(20),
PAIS VARCHAR(15) NOT NULL)
```
- (C)

```
CREATE TABLE Fornecedores
(CNPJ CHAR(14) NOT NULL,
NOME VARCHAR(20) NOT NULL,
PAIS VARCHAR(15))
```
- (D)

```
CREATE TABLE Fornecedores
(CNPJ CHAR(14),
NOME VARCHAR(20) NOT NULL,
PAIS VARCHAR(15) NOT NULL),
PRIMARY KEY (CNPJ)
```
- (E)

```
CREATE TABLE Fornecedores
(CNPJ INTEGER(14) NOT NULL,
NOME VARCHAR(20),
PAIS VARCHAR(15) NOT NULL),
PRIMARY KEY (CNPJ)
```

**46**

Para que fosse mais fácil entender um sistema em desenvolvimento, um desenvolvedor usou um modelo de dados relacional (protótipo) mais simples do que o do banco de dados do sistema corporativo original, sendo que ambos utilizavam o mesmo SGDB PostgreSQL. Cabe ressaltar que esse protótipo utilizava apenas um subconjunto dos dados do sistema corporativo original e realizava apenas consultas.

Uma forma de garantir que os dados desse protótipo estejam sempre completamente atualizados em relação aos dados reais, com baixo impacto tanto na operação quanto no desempenho do sistema corporativo original, é

- (A) criar apenas VIEWS no protótipo, definidas com consultas sobre as tabelas do sistema corporativo original.
- (B) implantar TRIGGERS a cada INSERT, em todas as tabelas do sistema corporativo original, atualizando as tabelas do protótipo.
- (C) implantar TRIGGERS de atualização a cada SELECT, em todas as tabelas do protótipo.
- (D) particionar as tabelas da base do sistema corporativo original escolhendo um RANGE adequado ao trabalho do protótipo.
- (E) utilizar DUMP da base do sistema corporativo original e PSQL para a base do protótipo, a cada seção de trabalho, para atualizar a base do protótipo.

49

bd

Ao desenvolver um Data Warehouse para o Banco W, um programador decidiu criar um modelo conceitual com base no modelo estrela para cada fato analisado. Ao criar a primeira tabela fato, relativa ao valor e ao prazo de empréstimos, foram identificadas as seguintes dimensões, com os seus atributos descritos em parênteses: tempo (dia, mês e ano), agência (estado, cidade, bairro e número da agência), produto (nome do produto e juros do produto) e cliente (conta e nome do cliente).

Segundo as regras e as práticas da modelagem dimensional, e usando a granularidade mais baixa, que atributos devem constar da tabela fato?

- (A) fato_id, dia, mes, ano, estado, cidade, bairro, numero_agencia, nome_produto, juros_mensais_produto, conta_cliente, nome_cliente, valor_emprestimo, prazo_emprestimo
 (B) fato_id, emprestimo_id, valor_emprestimo, prazo_emprestimo
 (C) fato_id, tempo_id, agencia_id, produto_id, cliente_id, emprestimo_id
 (D) fato_id, tempo_id, agencia_id, produto_id, cliente_id, dia, mes, ano, estado, cidade, bairro, numero_agencia, nome_produto, juros_mensais_produto, conta_cliente,

52

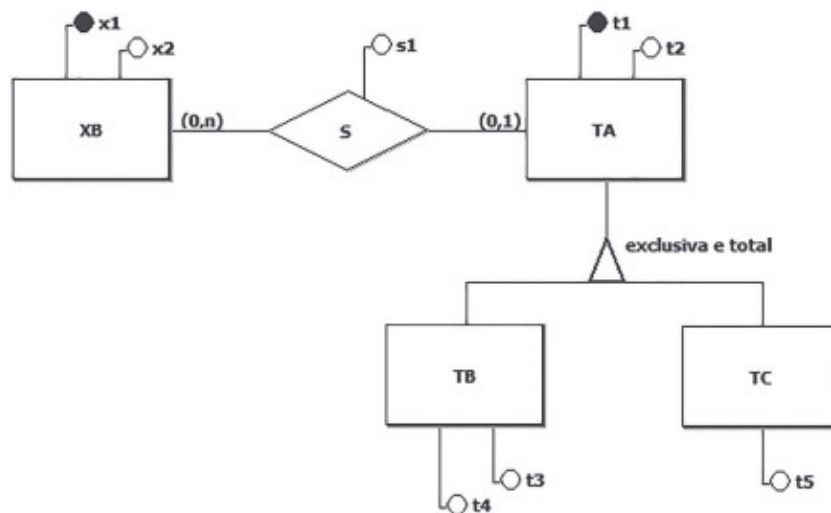
bd

Esquemas de bancos de dados relacionais podem ser descritos por meio de notações alternativas à linguagem SQL. Em uma dessas notações, uma tabela (relação) é descrita por meio de um nome e de um conjunto de nomes de colunas (atributos), separadas por vírgulas.

Os demais elementos de uma tabela são os seguintes:

- Colunas sublinhadas compõem a chave primária.
- Colunas que admitem o valor nulo são exibidas entre colchetes.
- Chaves estrangeiras são representadas por meio da cláusula REF: <lista_de_colunas> REF <nome_de_tabela>

Nesse contexto, considere a Figura a seguir, que exibe um diagrama E-R.



Qual esquema relacional preserva a semântica desse diagrama E-R, sem a necessidade de criação de regras de integridade adicionais?

- (A) XB(x1, x2, t1, s1)
t1 REF TA

TA(t1, t2, tipo, t3, t4, t5)

- (B) XB(x1, x2)

TA(t1, t2, tipo)

TB(t1, t3, t4)
t1 REF TA

TC(t1, t5)
t1 REF TA

S(x1, t1, s1)
x1 REF XB
t1 REF TA

- (C) XB(x1, x2)

TA(t1, t2, tipo, [t3], [t4], [t5])

S(x1, t1, s1)
x1 REF XB
t1 REF TA

- (D) XB(x1, x2)

TA(t1, t2, tipo, [t3], [t4], [t5])

S(x1, t1, s1)
x1 REF XB
t1 REF TA

- (E) XB(x1, x2, t1, s1)
t1 REF TA

TA(t1, t2, tipo)

TB(t1, t3, t4)
t1 REF TA

TC(t1, t5)
t1 REF TA

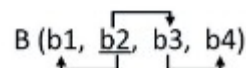
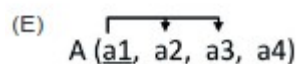
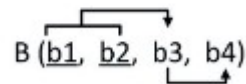
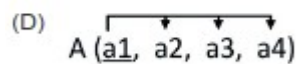
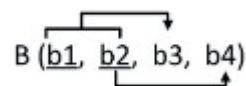
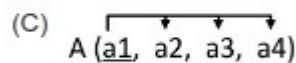
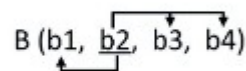
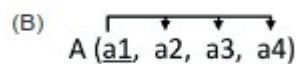
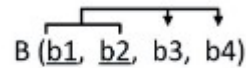
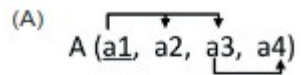
Continua



56

Na descrição de esquemas de banco de dados relacionais, a notação $A \rightarrow B$ indica que B depende funcionalmente de A (ou que A determina B).

Admitindo-se que todas as relações apresentadas a seguir atendem à 1FN, o único esquema que se encontra na 3FN é



65



Um banco comercial deseja obter um tipo de banco de dados NoSQL que trate os dados extraídos de redes sociais, de modo a formar uma coleção (collection) interconectada. Nessa coleção (collection), os dados são organizados em vértices ou objetos (O) e em relacionamentos, que são relações (R) ou arestas.

Nesse modelo de banco de dados NoSQL, os dados seriam apresentados da seguinte forma:

O:Usuario{u1:Joao, u2:Jose, u3:Maria, u4:Claudio}

O:Escola{e1:UFRJ, e2:URGS, e3:IFB}

R:Estudaem{re1=u1:e2;re2=u2:e2;re3=u3:e1;re4=u4:e3}

R:Amigode{ra1=u1:u2;ra2=u1:u3;ra3=u2:u3}

O banco de dados NoSQL que representa essa situação deve ter uma estrutura do tipo

(A) Distributed Hashing

(B) Consistent Hashing

(C) Document Oriented

(D) Graph Oriented

(E) Vector Clock

Um administrador de um banco de dados construído por meio do MongoDB inseriu dados em uma coleção (collection) de dados da seguinte forma:

```
db.fornecedores.insert( {  
  codigo: "thx1138",  
  nome: "Roupas Syfy Ltda",  
  pais: "Arabia Saudita" } )
```

Posteriormente, esse administrador construiu uma consulta que retornou apenas o nome, sem repetição, de todos os países que fazem parte dessa coleção (collection).

O comando utilizado para tal consulta foi

- (A) `db.fornecedores.find("pais")`
- (B) `db.fornecedores.find().pretty({"pais":1})`
- (C) `db.fornecedores.find().sort({"pais":1})`
- (D) `db.fornecedores.distinct({"pais":0})`
- (E) `db.fornecedores.distinct("pais")`